

FPGA IMPLEMENTATION OF VISUAL OBJECT TRACKING SYSTEM

Nishchay Malik
Roll No.: 213ec2204



Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela - 769 008, India

FPGA IMPLEMENTATION OF VISUAL OBJECT TRACKING SYSTEM

A thesis submitted in the partial fulfilment of the requirements of the
Degree of

**Master of Technology
in**

VLSI DESIGN AND EMBEDDED SYSTEM

Submitted by

**Nishchay Malik
(Roll No: 213ec2204)**

Under the guidance of

Prof. KamalaKanta Mahapatra
Professor



Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela - 769 008, India
May 2015



Department of Electronics & Communication Engineering
NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA
ODISHA, INDIA – 769 008

CERTIFICATE

This is to certify that the thesis titled **“FPGA IMPLEMENTATION OF VISUAL OBJECT TRACKING SYSTEM”** submitted to the National Institute of Technology, Rourkela by **Nishchay Malik**, Roll No. **213EC2204** for the award of the degree of **Master of Technology in Electronics & Communication Engineering** with specialization in **“VLSI Design and Embedded Systems”**, is a bonafide record of research work carried out by him under my supervision and guidance. The candidate has fulfilled all the prescribed requirements.

The thesis, which is based on candidate's own work, neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere. To the best of my knowledge, the thesis is of standard required for the award of the degree of **Master of Technology** in Electronics & Communication Engineering.

Place: Rourkela

Prof. Kamalakanta Mahapatra

Department of Electronics &
Communication Engineering NATIONAL
INSTITUTE OF TECHNOLOGY
Rourkela-769 008 (INDIA)

ACKNOWLEDGEMENT

I am very grateful and would like to thank my supervisor **Prof. Kamalakanta Mahapatra** for his guidance, advice and constant support throughout my thesis work here in National Institute of Technology, Rourkela, without him it would not have been possible for me to complete my thesis.

I am highly grateful to all the faculty members and staff of the Department of Electronics and Communication Engineering, N.I.T. Rourkela for their unforgotten teaching and motivation in my research work.

Besides, my sincere thanks to Mr Vijay Kumar Sharma, Ph.D. Scholar ECE dept. for his technical support for the completion of my project. It is he who has helped me to learn the programming and optimizing skills for which I would be grateful to him throughout my life.

I would like to thank all my friends, classmates, lab mates for giving me their valuable advises and thoughtful ideas, which grew a greedy interest for me towards my research area.

I would like to express my gratitude to Tom sir, Jagannath sir, and sudheendra sir who has given the support in carrying out the work.

I would like to show my gratitude for my parents, for their sacrifice throughout my life. They are the inspiration of my research work.

NISHCHAY MALIK

Contents

Certificate	iii
Acknowledgement	iv
List of Figures	vii
List of Tables	ix
Abstract	x

Chapter 1

Introduction.....	1
1.1. Object Tracking system	2
1.1.1. Object Detection	3
1.1.2. Object Classification	4
1.1.3. Object tracking	5
1.2. Objective of The Work	6
1.3. Project Applications.....	7
1.4. Literature Review.....	7
1.5. Thesis Organization	8
1.6. Summary	9

Chapter 2

Tracking with kernels: MATLAB Simulation	10
2.1 Introduction.....	11
2.2 Learning in Vision	12
2.2.1 Component of learning	12
2.2.2 Models for learning.....	12
2.2.3 Kernel methods	14
2.3 Regularization Risk	14
2.4 Circulant Matrix	15
2.5 Tracking-by-detection with kernel	16
2.6 MATLAB Results	20
2.7 Summary	23

Chapter 3

Hardware Implementation	24
3.1 FPGA Introduction	25

3.1.1	Configurable Logic Block	27
3.1.2	Interconnect matrix and IOBs	27
3.2	FFT Module	29
3.2.1	R2SDF algorithm	30
3.2.2	RTL Block and Simulation results.....	33
3.3	FFT in 2D.....	37
3.4	Implementation of Cordic Algorithm	38
3.4.1	RTL Block and Simulation results.....	39
3.5	Implementation of Gaussian Kernel Classifier Module.....	40
3.5.1	RTL Block and Simulation results.....	41
3.6	Implementation of VGA Controller	42
3.6.1	Timing Diagram	42
3.6.2	RTL Bock and Simulation results.....	43
3.7	Summary	45
Chapter 4		
Conclusion and Future Scope		46
4.1	Conclusion	47
4.2	Future work	47
Bibliography		48

List of Figures

Figure 1.1 Different steps involved in tracking system.....	3
Figure 1.2 Different classification of tracking methods.....	5
Figure 2.1 Linear SVM.....	13
Figure 2.2 Overfitting in machine learning algorithms	15
Figure 2.3 nxn Circulant matrix	16
Figure 2.4 Different sampling methods in tracking	17
Figure 2.5 Flow chart of tracking-by-detection.....	18
Figure 2.6 object sample at initial position	20
Figure 2.7 tracked object in different frame	21
Figure 2.8 image sample and kernel matrix	22
Figure 2.9 presion plot of given algorithm for Gaussian kernel.....	23
Figure 3.1 typical architecture of FPGA integrated circuit	25
Figure 3.2 EP2C20 cyclone II family device block diagram	26
Figure 3.3 CLB architecture	27
Figure 3.4 Interconnect fabric in FPGA	28
Figure 3.5 IOBs circuit in FPGA	28
Figure 3.6 8-point DIF FFT signal flow graph	31
Figure 3.7 R2SDF stage block diagram	32
Figure 3.8 1024/512/256/128/64 point FFT Block Diagram for R2SDF	33
Figure 3.9 TOP RTL block of 8-point FFT.....	34
Figure 3.10 Controllor and data path block diagram for 8-point FFT.....	34
Figure 3.11 8-point FFT stage RTL circuit	35
Figure 3.12 R2SDF FFT stage RTL Circuit diagram.....	35
Figure 3.13 8-point FFT simulation result	36
Figure 3.14 MATLAB result of 8-ponit FFT	36
Figure 3.15 1024/512/256/128/64 point FFT RTL datapath block	36
Figure 3.16 vector rotation in a unit radius circle.....	38
Figure 3.17 Cordic top level RTL block	40
Figure 3.18 Cordic simulation result	40
Figure 3.19 RTL Block of Cordic for Exponential function calculation.....	41
Figure 3.20 simulation result of Cordic For exponential function	41

Figure 3.21 VGA timing diagram	42
Figure 3.22 VGA controller RTL block	43
Figure 3.23 simulation result of VGA controller	44

List of Tables

Table 1.1 Comparison of various object detection methods	4
Table 3.1 variable FFT Macro statistics	36
Table 3.2 HDL Synthesis report of Cordic macro statistics	39
Table 3.3 Device utilization summary of different module.....	41

ABSTRACT

Object tracking is a crucial problem of computer vision, due to the high dimensional data processing, which required huge memory resource to hold the intermediate image data. Due to complexity in real time object tracking, this field is very challenging for researchers. In today's world technology high speed and low power computers, and cheap and high definition cameras availability increase the interest of researcher in object tracking systems. In computer vision task real time object tracking is becoming challenging ingredient.

Here a new approach for real time object tracking is taking place to implement on a reconfigurable hardware. Reconfigurable hardware is best target hardware for video processing algorithms, because they use parallel processing in contrast to conventional processor, which they process data serially. A novel video processing algorithm for tracing object in consecutive frames is implemented here on the FPGA based hardware. The algorithm is based on machine learning's algorithm of discriminative classifier. The central objective of my work is to track object in real time with less hardware complexity. The FPGA based VLSI architecture for tracking-by-detection based object tracking system was implemented and verified in Altera Quartus II tool and FPGA board. The target device for reconfigurable hardware was Altera cyclone II EP2C35F672C6 FPGA.

The bottleneck in this architecture is FFT, because FFT is the main block which is used repetitively in the hardware. Hence to emphasized hardware pipelined single delay feedback FFT block with Cordic was implemented. Cordic was used for twiddle factor and exponential function calculation.

CHAPTER 1

INTRODUCTION

- Object Tracking System
- Objective of The Work
- Project Applications
- Literature Review
- Thesis Organization
- Summary

1.1 Object Tracking System

Object tracking has significant role in computer vision task, the key objective of computer vision is to replicate the human vision functionality such as motion perception and scene understanding. One of the critical challenge in object tracking is real time speed requirement [1]. Computer vision has wide range of application in imbedded systems, such as computer human interaction, which can be very useful in autonoums vehicle navigation, surveillance, and robotic intelligence [2]. Like other computer vision task, to extract required information from scene object tracking used intensive computational resources. There are so many factors which affect the real time object tracking and making its complex to track the object, partial or full occlusion of object, abrupt motion, and rigid-object in scene these are the most encountered in tracking. A typical video tracker has two major parts, target representation, and localization. Target representation deals with classification of objects, such as human, cars, animals, whereas localization is used to maximize the likelihood function.

Following are the basic steps involved in object tracking system [2]

- Object detection
- Object classification
- Object tracking

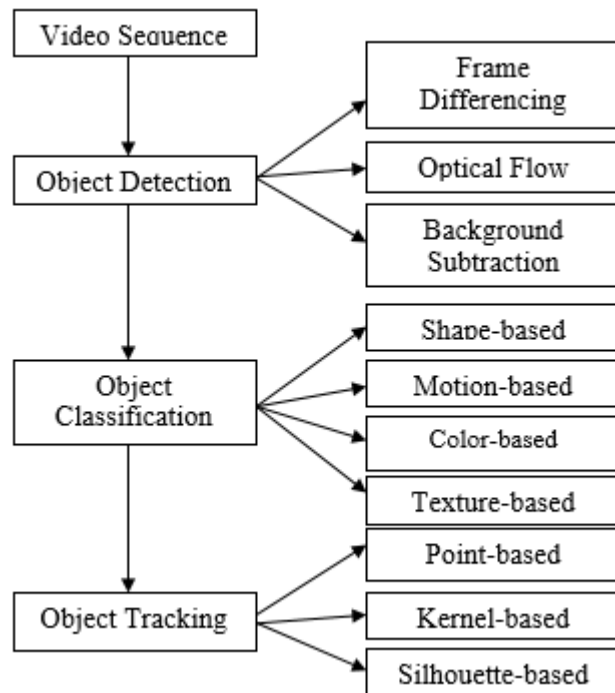


Figure 1.1 Different steps involved in tracking system [2]

1.1.1 Object Detection

The first step in tracking system is to identify the object of interest in a video and segment that object from scene. Object detection can be achieved by various methods such as Frame differencing, Optical Flow, Background subtraction. Frame differencing method is very easy because in this method only pixel to pixel subtraction operation is involved this is not an efficient method, in contrast background subtraction is the efficient method, in which by processing pervious frames a background model is used, so no background is required as in frame differencing. Optical Flow method gives you the full movement information of interest object. In many object detection method temporal information is used to detect object, this temporal information reduce the false detection. Object detection

can be done by frame differencing or by background subtraction for which a model of background is required.

TABLE 1.1

Comparison of various object detection methods [2]

Methodology	Accuracy	Timing	Merits	Demerits
Background Subtraction	Moderate	Moderate	Low memory required	Ineffective in multimodal background
Optical Flow	Moderate	Moderate	Gives complete movement info	Calculation is large
Frame Differencing	High	Low	Easiest Method	Background required

1.1.2 Object classification

The interest object can be human, vehicle, and animals, to identify these object used object classification based on their respective features. There are many features can be used for object classification, motion, shape, Color, and texture [2].

- Shape-based method: Diverse depictions of shape data of movement districts, for example representations of pixel points, box and blob are accessible for characterizing moving articles.
- Motion-based method: motion is the key feature of any object. Optical Flow, Residual Flow can be used to model the rigidity and motion of object.
- Color-based method: Like other numerous image feature color feature is relatively steady under perspective change.
- Texture-based method: Texture method checks the gradient orientation of local image region. Local contrast normalization can be used to improve efficiency of method.

1.1.3 Object tracking

Tracking is defined as the trace of an object in a scene, the purpose of tracking method is to get the track of object over time in all frame of video. Object tracking is done for object segmentation, object recognition and decision making about activities. Object tracker provide the region in mage, which is occupied by object. As describe in [3], tracking can be divide into following classes:

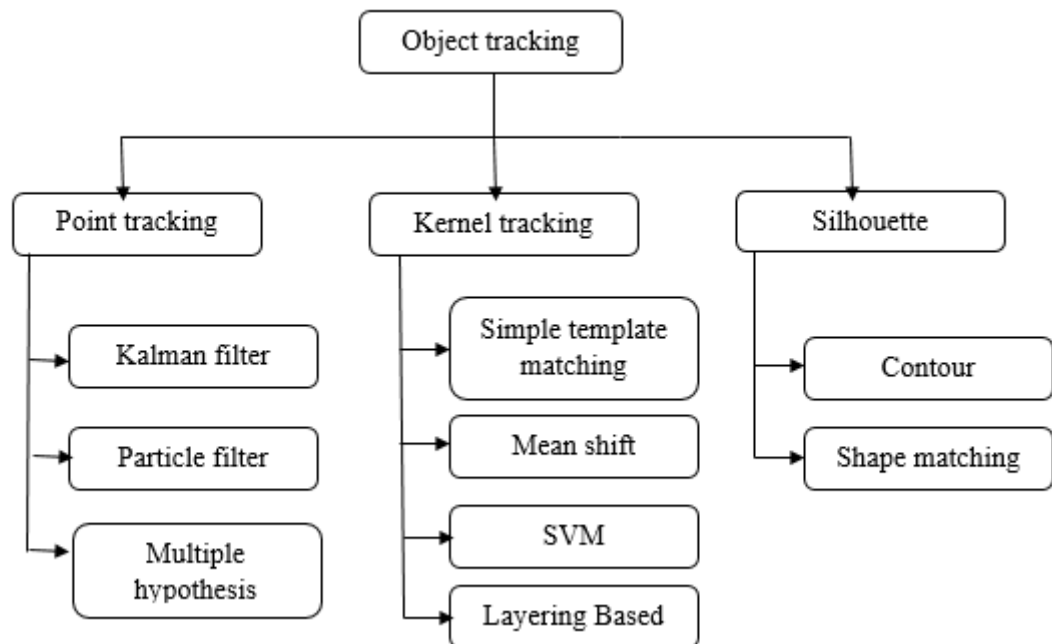


Figure 1.2 Different Classification of tracking methods [2]

Point tracing: In this approach detected object is represented by set of points. Particle filtering, Kalman filter is the most famous methods in point tracking.

Kernel Tracking: This approach is based on appearance model, the main goal of this approach is to estimate object motion.

Silhouette Tracking: In this method information, which is encoded into the contour of object region. Complex shape are describe by silhouette accurately. In silhouette tracker occlusion of object is the main aspect, in this method occlusion problem not to explicitly define.

1.2 Objective of the work

After studying literature on object tracking, found that the tracking in real time is very challenging task. Automatic tracking is the foundation of many computer vision application.

The objective of my work is as follows:

- To implement and simulate tracking-by-detection exploiting Circulant matrix algorithm in MATLAB.
- To implement an efficient object tracking algorithm on FPGA chip, which track interest object in a sequence of frames
- Along with implementation on FPGA, computation time, functional verification, silicon area and power consumption of system also calculated.
- Implemented algorithm should be work in real time, and accurately track the object.
- Initially custom hardware for algorithm is implemented and later it interfacing with other hardware to interact with outer world.
- Design software portion for the processor core for controlling all hardware modules.
- All required hardware subsystem for algorithm was implemented, verified, and compared with other hardware.

1.3 Project Application

Object tracking has many applications, increasing with time. Some applications listed as below

- Visual surveillance system
- Autonomously vehicle navigation
- Automatic traffic control
- Medical imaging
- Intelligent robotics
- Human machine interaction
- Video compression
- Video indexing
- Augmented reality

1.4 Literature Review

- Jean Gao et al. [7] In this paper a new approach on motion-estimation has been used for tracking object. In this approach no camera calibration, movement in camera has no effect, and this approach comes under semi-automatic class. This approach is based on Kalman-filter, motion parameters are updated using recursive Kalman-based updating. In this approach human help in initial frame for segmentation of object is required to track the object in further frames, it is feature-correspondence approach.
- Sheikh Nasrullah, Dawood Ashraf [8] In this paper Based on Block Matching Algorithm, motion model a new approach for search strategy has been taken for reducing complexity of computer resources. New object feature block search

method reduce the search are and reduce the computer complexity, which is the key problem in real time tracking.

- Jung UK Cho, Seung Hun Jin, Xuan Dia Pham, et al.[9] in this paper author used color feature for tracking object. In this paper probabilistic approach has taken for tracking, which is robust in terms of computational cost. Color histogram is used in this paper for modelling probabilistic model of color object.
- David S.Bolme, J.Ross Beveridge,[11] In this paper adaptive correlation filters is used to modelled appearance of object, and convolution is used to track object. This approach is invariant to the change in illumination, scale and non-rigid object deformation
- Boris Babenko, Ming-Hsuan Yang, Serge Belongie,[12] In this paper machine learning method are used, which is based on discriminative classifier. This is a class of tracking method called tracking-by-detection, which gives real time tracking results. In this paper a new learning algorithm is used, it called multiple instance learning.

1.5 Thesis organization

This Thesis is organized as follow

Chapter-1: In this chapter an introduction of tracking system. A brief overview of different subsystems required for tracking system.

Chapter-2: In this chapter an introduction of tracking-by-detection methods and some Machine Learning techniques are discussed. MATLAB Implementation and simulation of Discriminative classifier based tracking algorithm also presented.

Chapter-3: In this chapter implementation and verification of various hardware module for tracking system is presented. Variable point FFT, Cordic, Gaussian classifier, and VGA controller is implemented in Verilog HDL. The simulation result also discussed.

Chapter-4: Conclusion and future work of the project is drawn here.

1.6 Summary

Object tracking has numerous application in the field of computer vision, robotic vision etc., to develop intelligence machine a real time object tracking algorithm required. In this chapter a brief introduction of object tracking and different steps involved in object tracking systems. Following issues regarding to object tracking summarizes as follow

- There are some difficulties in reliable tracing of object, such as rapid change in appearance due to change in image noise, illumination, and non-rigid motion.
- In tracking methods many kind of object features are used such as corners, edges, motion, and color.
- There are many tracking algorithms, which have its own advantages and disadvantages.
- Real time speed is the main requirement in object tracking.

Chapter 2

Tracking with kernels: MATLAB Simulation

- Introduction
- Learning in vision
- Regularization
- Circulant matrix
- Tracking-by-detection with kernel
- MATLAB Results
- Summary

2.1 Introduction

The past few years have witnessed increased interest in the usage of machine learning algorithms within the object tracking systems, in these algorithms samples of object collected online to train the classifier. To train classifier large samples may conflict real time tracking requirement, in contrast limiting the samples may degrade the performance of tracker [3]. Recent work in a class of tracking methods called tracking-by-detection has been indicated to give promising results in real time environment. In some application the interest object is known to the tracker in advance, this prior knowledge of object can be utilize in tracking methods. In other cases, tracking of arbitrary objects, that is specified at current time of tracking. In computer vision, tracking difficulty rely upon several aspects, for example the prior knowledge of interest object and number of object to track. Tracking of generic objects are the challenge for researchers because it's abrupt change in appearance due to rotating in the plane, illumination change and deforming, has remained. A tracking system has three key aspect: (1) appearance model; (2) motion model; (3) search strategy for likely location of object. In tracking-by-detection approach tracking problem treated as a classification of object by use of online learning methods to update the tracking model. In tracking-by-detection, classifier is used to find object location by Regularized risk minimization.

2.2 Learning in vision

Computer vision treat as machine learning problem, in pattern recognition, based on prior knowledge of image data we infer new image. This process divided into two parts: first one is learning in which we display the relationship between the image information and scene content. The second one is inference, in which we exploit this relationship to predict parameter for new images [5]. If the infer state is continuous it called regression other hand for discrete state we call it classification.

2.2.1. Component of Learning

Learning has following component to solve vision problem

- A *model* which relate the image data and state parameter mathematically. This model specifies all possible relationship between sample image and its output label. Relationship is determined by the model parameters.
- A *learning* algorithm that fit the model parameter with least risk error to training data which is the paired set of image data and its state.
- An *inference function* that takes new image data, and according to defined model return the sate parameter.

2.2.2. Models for learning method

There are two models used for modelling relationship between sample data and the assigned state of data.

- **Discriminative Model:** It is a conditional model, used in machine learning in a probabilistic framework, if unobserved variable is y and observed variable is x then dependence of y on x , conditional probability is used to prediction of y from x it is denoted by the probability distribution function $P(y|x)$.

Discriminative models come under supervised learning, logistic regression, SVM, linear regression, neural network these are the examples of discriminative models.

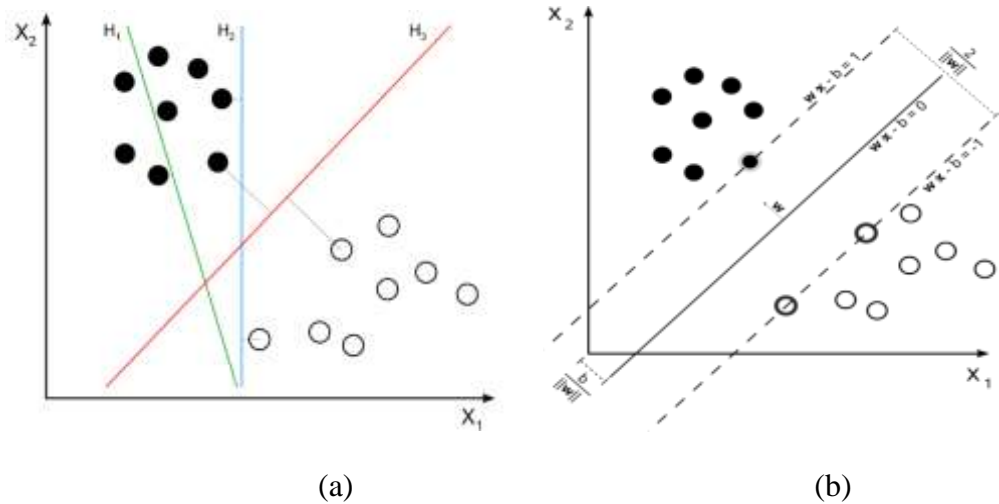


Figure 2.1 Linear SVM [6]

Suppose we have two class of data denoted by black dot and white dot, the task of learning algorithm is to decide the class of new data. In SVM, classes are separated by hyper planes there are many hyperplane to classify that data but best hyperplane is that which gives maximum margin to the separated classes, as shown in fig. 2.1 (a) plane H_3 classified data with maximum margin in compare to H_1 , H_2 .

- Generative Model:** In statistics modelling of randomly generated data point, with some unrevealed parameters done by generative models. Generative models incorporate joint probability distribution, and conditional distribution is done in generative model by using Bayes' rule. Generative models come under unsupervised learning where machine is trained by itself, Gaussian mixture model, Hidden Markov model etc. are generative models examples.

2.2.3 Kernel methods

Kernel methods are a class of machine learning algorithm, which based on the inner dot product of instance. Kernel methods operate in high dimensional space, it is instance-based learner in contrast to learning some fixed parameter of set of input features. In kernel trick prediction of unlabelled inputs X' , is done by a similarity function. For binary classifier predicted output of new data is computed by weighted sum of kernel function given as

$$\hat{y} = \text{sgn} \sum_{i=1}^m w_i y_i k(x_i, x')$$

$\hat{y} \in \{-1, +\}$ is the classifier's predicted label of new input data, where y is the true output of data.

$k: \chi \times \chi \rightarrow \mathbb{R}$ is the kernel function denoted by $k(X_i, X')$ which calculate the similarity between any pair of input data $x, x' \in \chi$ where χ is the feature space of data sample, X is the training sample space. Linear Kernel function satisfied inner dot product of data set given as follow.

$$k(x, x') = \langle x, x' \rangle = \sum_i x_i x'_i$$

To learn nonlear data set kernel is used as a linear classifier in which mapping is done implecitly. There are some popular kernels such as Polynomial kernel, RBF kernel, Graph kernel used in computer vision.

2.3 Regularization Risk

Overfitting is the key problem in machine learning. In machine learning algorithms function is inferring from the training data, goal of this inferring function is to fine a function that most accurately infer output from training data, but not to map most

closely fits the data. This overfitting problem gives unstable solution, small change in training data cause large drift in inferring function, generalization is require and this can be achived by regularization method. In this bellow fig. a learning algorithm example of overfitting

is taken. Suppose red dotes shown in fig represent training data and the green line show the actual function which infer data set, while blue line represent the inferred function by learning algorithm, which overfit on data.

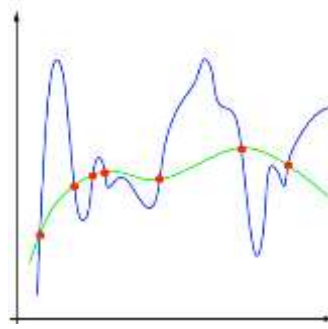


Figure 2.2 Overfitting in machine learning algorithms

The risk of hypothetical function is give as

$$R(h) = E[L(h(x), y)] = \int L(h(x), y) dP(x, y)$$

Where L is loss function, in general this risk cannot be calculated because distribution function is unknown, however we can only approximate the risk it's called Empirical risk minimization this can be calculated by taking average of loss function.

$$R_{emp}(h) = \frac{1}{m} \sum_{i=1}^m L(h(x_i), y_i)$$

2.4 Circulant Matrix

In mathematics Circulant matrix is a kind of toeplitz matrix, in which each row is rotating one element relative to the neighbour row. An nXn Circulant matrix given as

$$C = \begin{bmatrix} c_0 & c_{n-1} & \dots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \dots & c_1 & c_0 \end{bmatrix}$$

Figure 2.3 nxn Circulant matrix

There is no need to define Circulant matrix explicitly, it is specified by one vector, it can be first row or column, the remaining row or column can be defined by circular permutation of vector.

Circulant matrix have some properties, which can be exploit in kernel methods in machine learning algorithms. Geometrically Circulant matrix has a relation with discrete Fourier transform, Circulant matrix hold the convolution operator kernel for the vector of any row of the matrix. Convolution function is defined as follow

$$b = c * a$$

in terms of Circulant matrix this is define by multiplication of vector a and the Circulant matrix of c as the vector, which cyclic rotate given as

$$b_k = \sum_{i=0}^{n-1} a_i c_{k-i}$$

2.5 Tracking-by-Detection with kernel

In recent years Discriminative classifier is used in tracking systems, after their success in detection methods. In discriminative model, classifier is trained by the samples, which is collected during training period. Classifier is train with samples collected around interest object, large samples increase the memory requirement, if reduce the samples it increase the redundancy, which lower the performance of classifier. As we know by increasing samples classifier is highly structured,

Circulant matrix method can be applied to train this classifier, a link to Fourier transform of Circulant matrix exploit this method to fast learning of dense sampling. In this algorithm samples are collected at initial frame of video and Fast Fourier Transform allows to train classifier for all sub-window without taking all the samples explicitly, only one sample is required around object location.

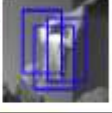
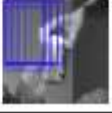
		Storage	Bottleneck	Speed
Random Sampling (p random subwindows)		Features from p subwindows	Learning algorithm (Struct. SVM [4], Boost [3, 6]...)	10 - 25 FPS
Dense Sampling (all subwindows, proposed method)		Features from one image	Fast Fourier Transform	320 FPS

Figure 2.4 different sampling methods in tracking [4]

In tracking-by-detection methods, actually it is an art of state, in which only initial location of object is inputting no other information is given. Based on initial location they collect sample data around the input initial location and classified the samples, if sample is near to the object assign positive label, otherwise make it negative[4].

Below the flow diagram of the algorithm is depicted

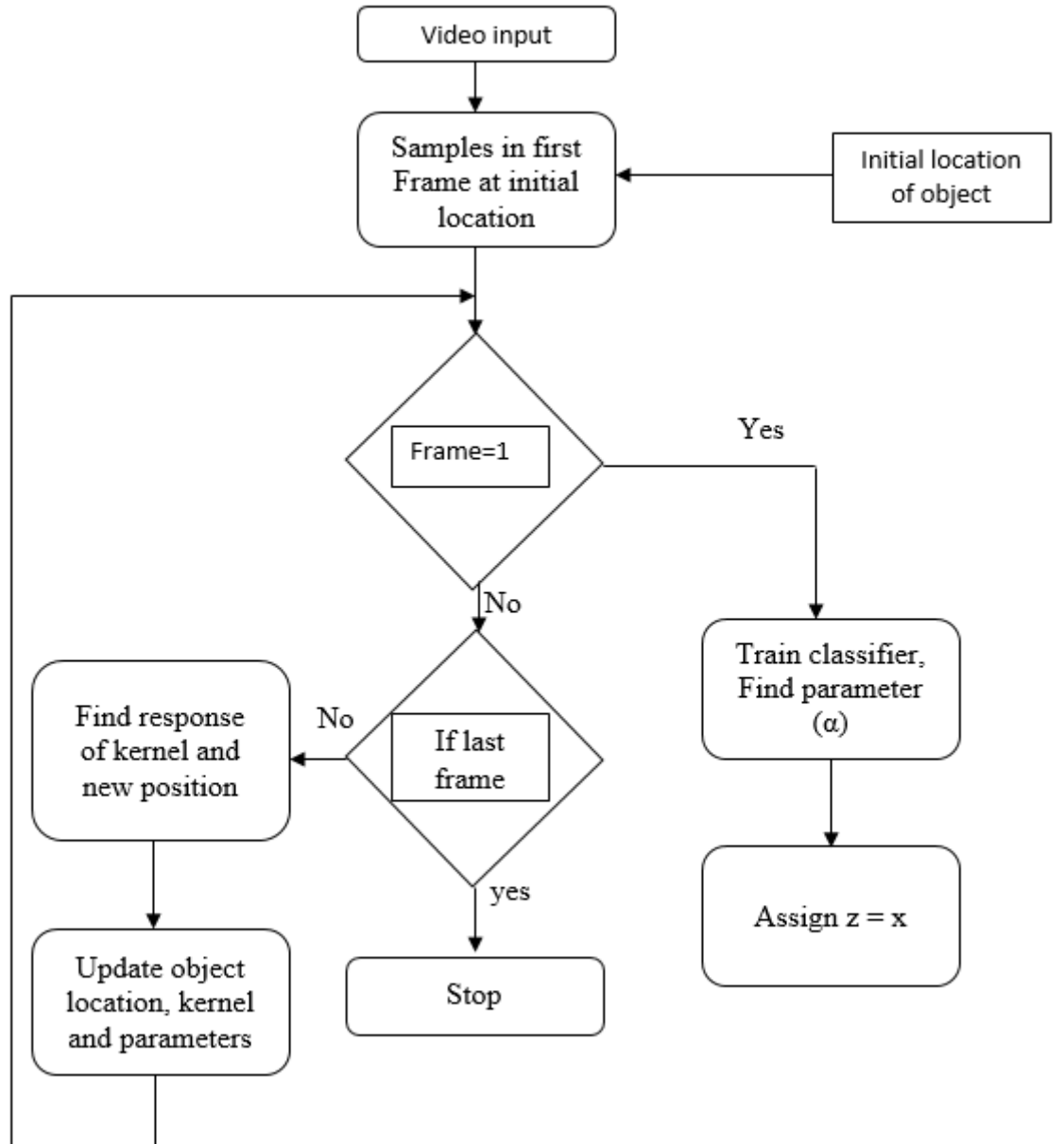


Figure 2.5 flow chat of tracking-by-detection algorithm

Methodology of the given algorithm, summarized as follow

- Discriminative model is the key point of algorithm, in the first frame of video the object sample is taken at given position and train classifier at all samples, resulting kernel classifier is highly structured. Consider liner Classifier, which has the form $f(z) = \langle a, z \rangle + b$, where $\langle ., . \rangle$ is a dot product, where $f(z)$ is the linear relating function defined by the

dot product of data set z and weight a minimization of regularization risk given as

$$\min_{a,b} \sum_{i=1}^n L(y_i, f(z_i)) + \lambda \|a\|^2$$

Where $L(y_i, f(z_i))$ is the loss function and λ is regularization parameter.

This can be applied at higher dimensional data by using kernel trick, which is given as follow

$k(z, z') = \langle \phi(z), \phi(z') \rangle$ where $\phi(z)$ is the feature space of samples.

Solution of classifier is defined as the liner combination of parameter and input mapped data [5].

$$W = \sum \alpha_i \phi(z_i)$$

Where α_i the parameter, with KRLS given as

$$\alpha = (K + \lambda I)^{-1} y$$

- In next step train classifier with all samples, called as dance sampling.

This can be achieved by Circulant matrix property dictated as below

If $C(a)$ is the Circulant matrix, defined by a . Then convolution of two vector a and b given as

$$C(u)v = F^{-1}(F^*(u) \Theta F(v))$$

Due to Fourier transform, only element wise product operation is carried out so training parameter is given as

$$\alpha = F^{-1}\left(\frac{F(y)}{F(k) + \lambda}\right)$$

- For training of classifier we take Gaussian kernel, which is the special case of Radial Basis Function (RBF) kernel. Describe as follows

It have the form for any function h given as

$k(z, z') = h(\|z - z'\|)$ where $\|z\|$ is the euclidian space

Because k is a Circulant matrix denoted as

$$K_i^{rbf} = k(z, p^i z') = h(\|z - p^i z'\|^2)$$

Expanding the norms, we get

$$k_i^{rbf} = h(\|z\|^2 + \|z'\|^2 - 2z^T p^i z')$$

By applied FFT, we get a general solution of RBF

$$k_i^{rbf} = h(\|z\|^2 + \|z'\|^2 - 2(F^{-1}(z) \Theta F^*(z')))$$

for Gaussian kernel h should be exponential function, given follow

$$k^{gauss} = \exp\left(-\frac{1}{\sigma^2}(\|z\|^2 + \|z'\|^2 - 2F^{-1}(F(z) \Theta F^*(z')))\right)$$

If we compute direct it takes $O(n^4)$ operation, if image is $n \times n$ size in frequency domain is reduced to $O(n^2 \log n)$ operations.

- Finally find the response of kernel with new image data. New position is given, where response is maximum. Response is given as

$$\hat{y} = F^{-1}(F(\bar{k}) \Theta F(\alpha))$$

2.6 Matlab results

For Matlab simulation, a standard video with 300 frames is used, to verify algorithm Gaussian kernel used for training of classifier. Here for Matlab simulation surfer video is considered, to get the result the first location is given by the user to initialize algorithm. In figure 2.6 first frame is taken, and green box represent the targeted object.



Figure 2.6 object sample at initial position



Figure 2.7 tracked object in different frame

Gaussian kernel is calculates as follow

Get the sample at current frame assign as z and increment frame then take sample of next frame at same position assign to x and calculate kernel function using below given formula

$$k^{\text{gauss}} = \exp \left(-\frac{1}{\sigma^2} (||x||^2 + ||z||^2 - 2F^{-1}(F(x) \Theta F^*(z))) \right)$$

-0.0302	-0.0032	0.0035	-0.0046	-0.0074	-0.0119	-0.0234
-0.0620	-0.0461	-0.0498	-0.0578	-0.0531	-0.0436	-0.0456
-0.1083	-0.1038	-0.1145	-0.1165	-0.0993	-0.0748	-0.0600
-0.1530	-0.1530	-0.1673	-0.1615	-0.1347	-0.1012	-0.0611
-0.1671	-0.1500	-0.1498	-0.1444	-0.1084	-0.0678	-0.0442
-0.1610	-0.1296	-0.1149	-0.1064	-0.0855	-0.0588	-0.0407
-0.1525	-0.0992	-0.0632	-0.0503	-0.0506	-0.0464	-0.0394

X

-0.0426	-0.0245	-0.0019	-0.0096	-0.0172	-0.0247	-0.0320
-0.0882	-0.0743	-0.0558	-0.0638	-0.0793	-0.0905	-0.0896
-0.1382	-0.1287	-0.1106	-0.1229	-0.1385	-0.1456	-0.1211
-0.1768	-0.1640	-0.1462	-0.1469	-0.1547	-0.1423	-0.1023
-0.1882	-0.1717	-0.1462	-0.1234	-0.1156	-0.0955	-0.0560
-0.1836	-0.1556	-0.1261	-0.0761	-0.0683	-0.0485	-0.0288
-0.1708	-0.1391	-0.1059	-0.0484	-0.0406	-0.0289	-0.0248

Z

0.8798	0.8806	0.8814	0.8821	0.8827	0.8832	0.8836
0.8804	0.8811	0.8817	0.8823	0.8828	0.8831	0.8833
0.8810	0.8816	0.8821	0.8825	0.8828	0.8830	0.8830
0.8817	0.8822	0.8825	0.8827	0.8829	0.8829	0.8827
0.8824	0.8827	0.8829	0.8830	0.8830	0.8828	0.8825
0.8832	0.8833	0.8834	0.8833	0.8831	0.8827	0.8823
0.8839	0.8840	0.8839	0.8836	0.8832	0.8827	0.8821

Kernel

Figure 2.8 image samples and kernel matrix

Precision plot: Precision plot is used to compare algorithm success in different sequences, means it tells how effective the algorithm is in given sequence. Here only given algorithm precision plot is drawn with ground data provided by the author [4]. Precision plot is drawn between the percentage of detected frame within the given ground truth threshold. Precision should be high in low threshold for best results. As the threshold increases, the graph shows that precision also increases. For good tracking algorithm, detected location in each frame should be near to ground truth, and threshold gives you how much the detected location of object is far from actual location in that frame, which is given in ground truth. Ground truth is actual locations of the object in each frame.

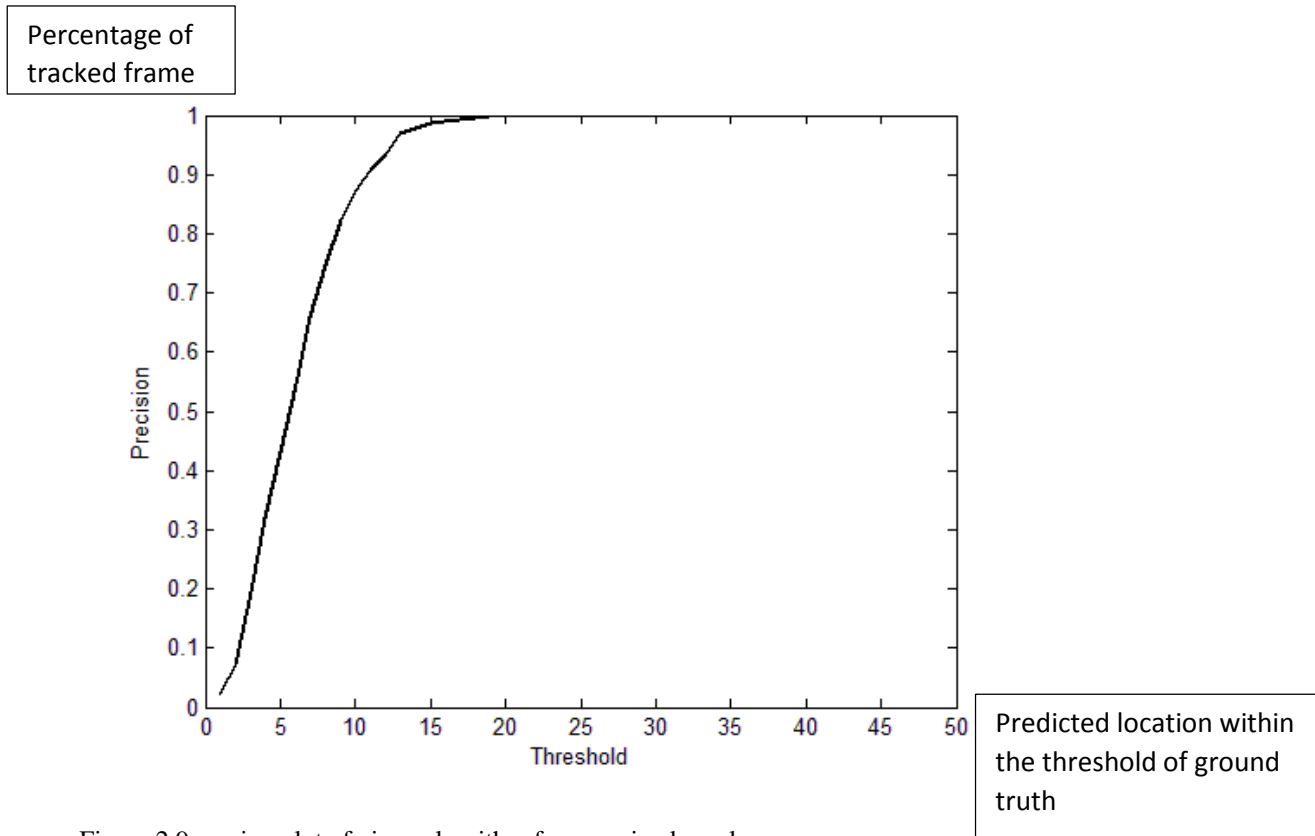


Figure 2.9 presion plot of given algorithm for guassian kernel

2.7 Summary

In this chapter MATLAB implementation and simulation of “tracking-by-detection Exploiting Circulant structure” algorithm, based on discriminative classifier has been done successfully. Precision plot is given here for comparing the result with other state of arts. The simulation result shows that algorithm work in real time speed, with less computational cost.

Chapter 3

Hardware Implementation

- FPGA Introduction
- FFT Module
- FFT in 2D
- Implementation of Cordic Algorithm
- Implementation of Gaussian Classifier Module
- Implementation of VGA Controller
- Summary

3.1 FPGA Introduction

FPGA is a programmable integrated circuit used for digital circuit design, FPGA contain logic blocks which can be configured by end user and connected with other logic block by switch matrix. FPGA is acronym for field programmable gate array, because interconnects are field programmable. In FPGA digital circuits are designed by HDL (Hardware Description Language) programming language which is specialized for hardware design. The performance of Digital circuits designed in FPGA is depends on the CLBs (complex logic blocks) architecture and the programming of interconnects. A typical FPGA architecture consist programmable CLBs, IOBs, and interconnects. Reconfiguration of FPGA makes it most choice for designer for prototyping digital circuits. Like other PLDs FPGA is also needed CAD tool for synthesis procedure.

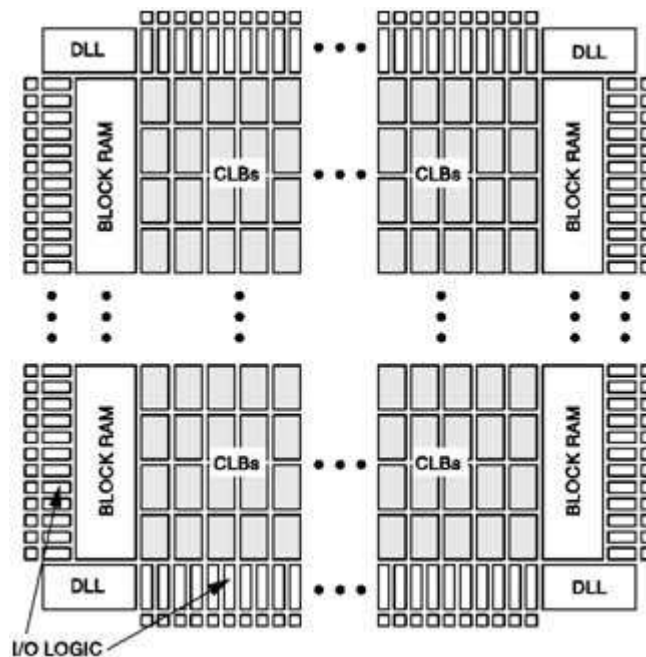


Figure 3.1 a typical architecture of FPGA integrated circuit

To implement complex digital circuit FPGA have reconfigurable logic elements, block RAM and for clock management they have DLL and PLL circuits. In some FPGA analog features are also added besides digital, such as programmable slew

rate. For clock circuit FPGA have dedicated clock routing network for regional and global clock. In addition, FPGA have analog PLL and DLL to synthesize different frequency and reduce clock jitters. FPGA have flexibility to implement soft processor cores and customized that according to your need. FPGA provide parallel processing of data, in contrast to ASIC where configurability done in software, hardware is fixed. Some feature of Altera FPGA EP2C35F672C6 listed as follow.

- It support 687K logic elements
- It contain 1.1 Mbits embedded RAM
- 18 x 18 embedded multiplier for DSP applications
- M4K memory block can be access as true dual port or single port
- Advanced I/O
- High speed interface for external memory
- Four PLL for Clock management

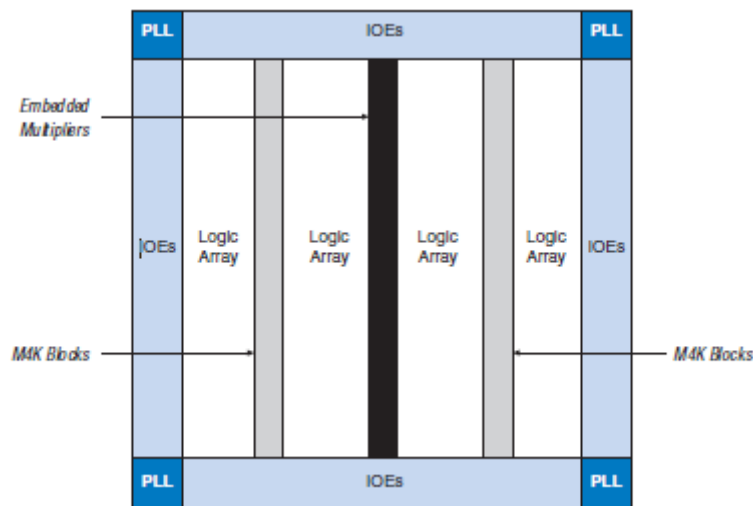


Figure 3.2 EP2C20 cyclone II family device block diagram

3.1.1 Configurable Logic Block (CLB)

CLB provides the end user configuration ability, each block have programmable RAM lookup table, Flip-Flops, multiplexers. Flip-Flop in CLB can be configured as edge-triggered. LUT are used to implement different function, and together they can be used as distributed RAM. A typical architecture of CLB shown in fig bellow

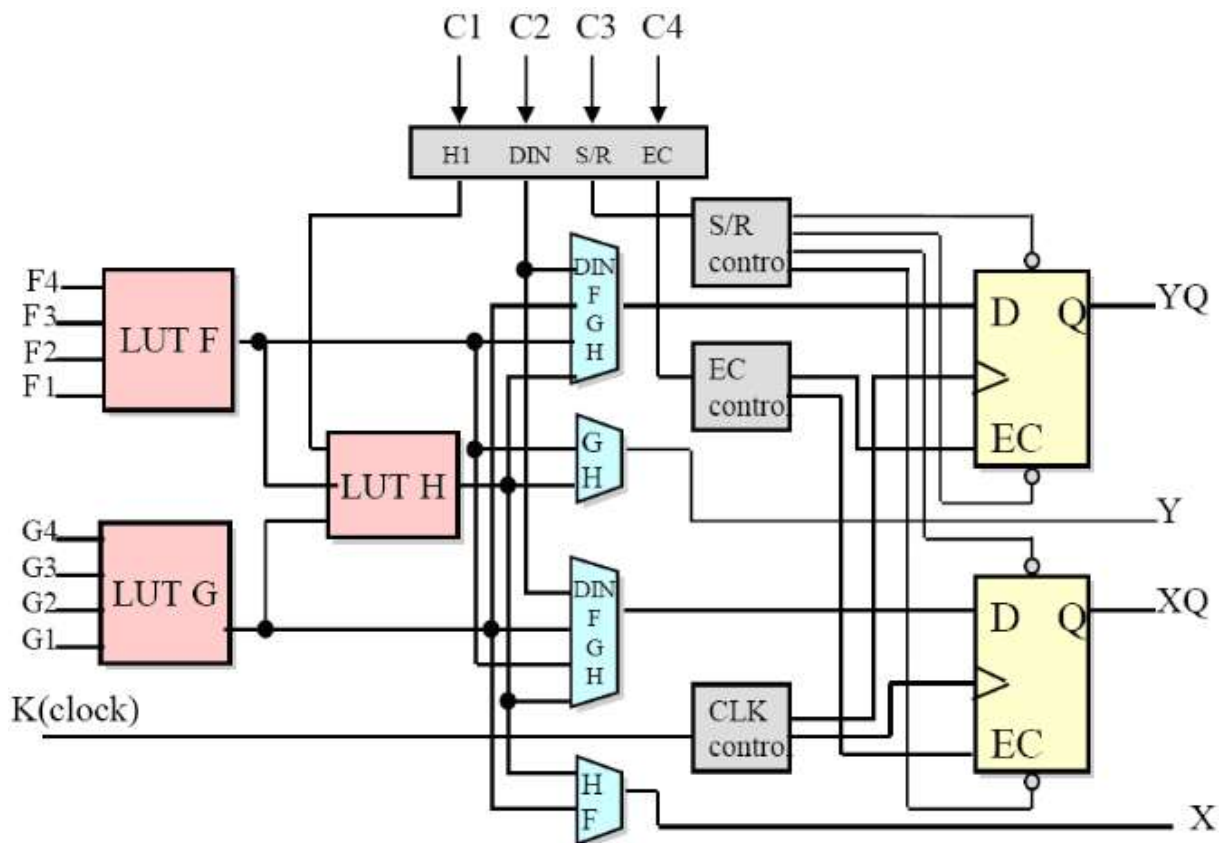


Figure 3.3 CLB architecture

3.1.2 Interconnect matrix and IOBs

To connect CLBs in FPGA used a switch matrix, direct interconnect path make connection between same column and row CLBs. To connect FPGA to outside world programmable IOBs are used, IOBs have buffers and it have compatibility with TTL and CMOS voltage levels. IOBs can be used as input port, output port, or bidirectional port, and connection can be direct, latched type.

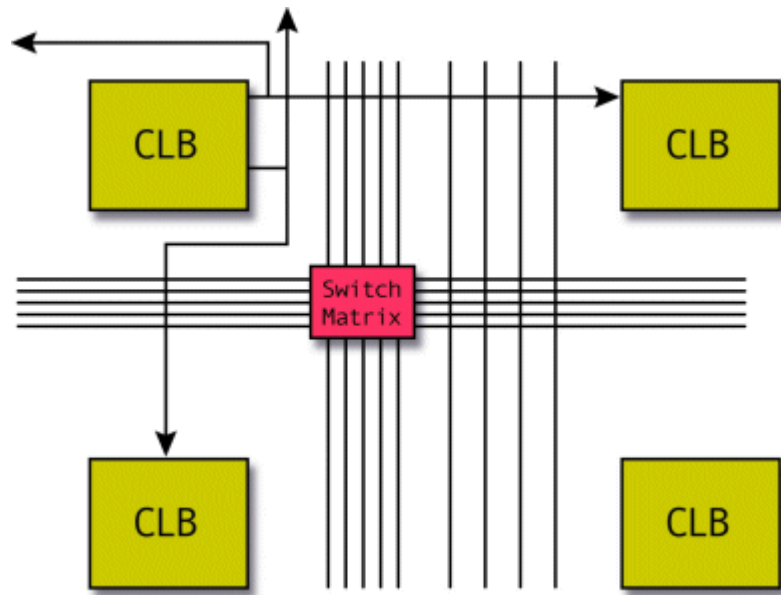


Figure 3.4 Interconnect fabric in FPGA

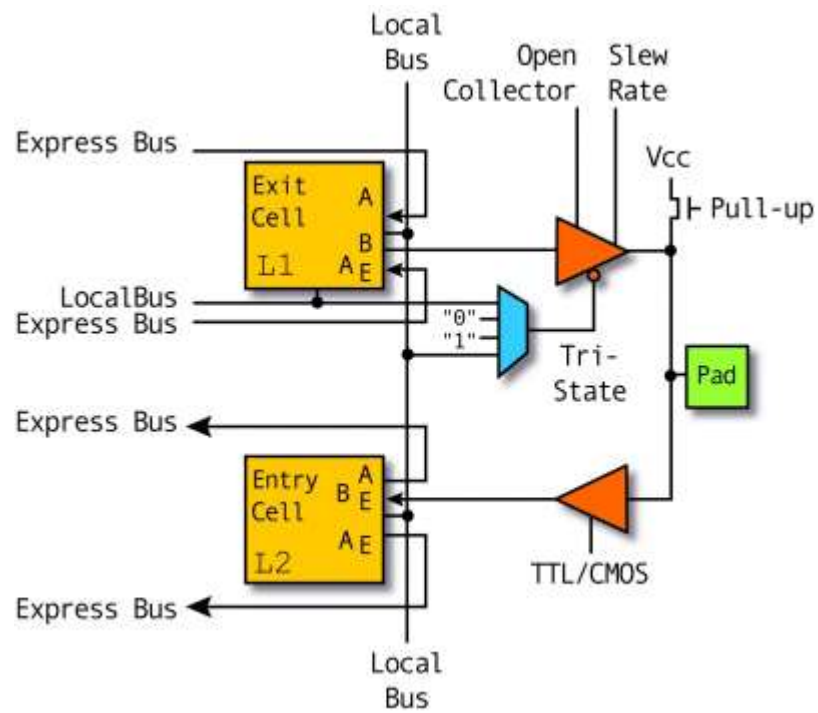


Figure 3.5 IOBs circuit in FPGA

3.2 FFT Module

FFT/IFFT is an algorithm, which is used to calculate Discrete Fourier transform (DFT) or Discrete Fourier transform (IDFT) of a series. DFT/IDFT of an N numbers sequence given as

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-\frac{2\pi kn}{N}}$$
$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{\frac{2\pi kn}{N}}$$

Where $e^{-\frac{2\pi kn}{N}} = w_N^{nk}$ is called twiddle factor also known as roots of unity, and $x(n)$ is the input sequence $X(k)$ is resulting Fourier sequence, in hardware implementation twiddle factor is pre-calculated and store in ROM. FFT algorithm gives fast calculation of DFT of the sequence with fewer resources.

Actual DFT has $O(n^2)$ Complexity FFT algorithm reduced to $O(n \log n)$. FFT algorithm attributed by Cooley and Turkey in 1965, different algorithm can be used to calculate FFT give as follow

- Decimation in time
- Decimation in frequency
- Prime factor algorithm
- Bluestein approach
- Bruun algorithm
- Rader algorithm

FFT algorithm used in many field of science, mathematics, communication, image processing etc. FFT can be implemented in pipelined, serial, and parallel architecture. Here pipelined architecture of FFT considered, pipelined FFT gives

high throughput in contrast to serial FFT. Pipelined FFT can be summarized as follow

- Radix-2 multipath delay commutator (R2MDC)
- Radix-2 single-path delay feedback (R2SDF)
- Delay buffer implementation
- Radix-4 algorithms

3.2.1 R2SDF algorithm

The signal flow graph of 8-point FFT is shown in figure below, this flow graph is based on DIF FFT algorithm for N point FFT stages given by $\log_2(N)$. 8-point FFT have 3 stages, in R2SDF algorithm calculation is done at each stage concurrently, so it increase the throughput. $N/2$ point divided into $N/4$ this process recursively applied until you get simple 2 point DFT which is the single butterfly computation.

Butterfly structure in FFT is the basic module for radix-2 Butterfly gives 2-point FFT. IFFT can be calculated with small change in FFT algorithm, twiddle factor have opposite sign. In R2SDF algorithm $(N/2^i)$ input is feed to the FIFO, where I is the stage of DIF FFT.

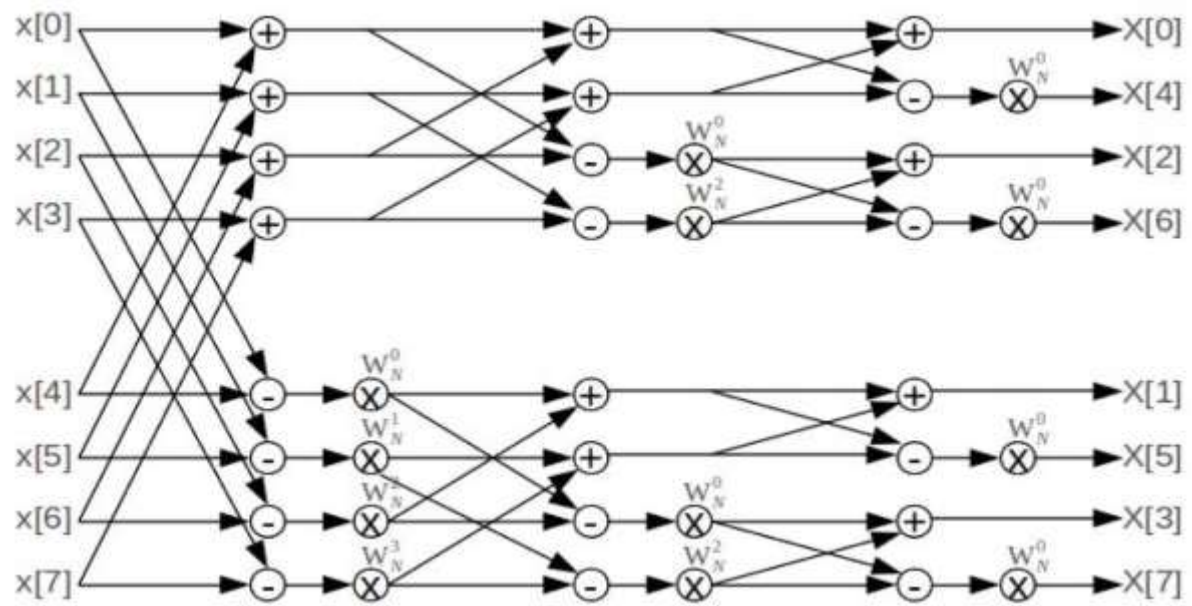


Figure 3.6 8-point DIF FFT signal flow graph

In R2SDF method first $N/2$ points are saved in FIFO then next phase $N/2$ point is fed with FIFO output to Radix-2 butterfly, which gives two complex output second output feed-back to the FIFO and first output of radix-2 butterfly gives to the next stage [14]. In next stage $N/2$ point of first stage feed, which again divide into two equal part and repeated same process of previous stage. At the time of stage second calculation stage one accept new sequence of input this pipelined structure improve the speed of FFT hardware, in contrast to serial FFT.

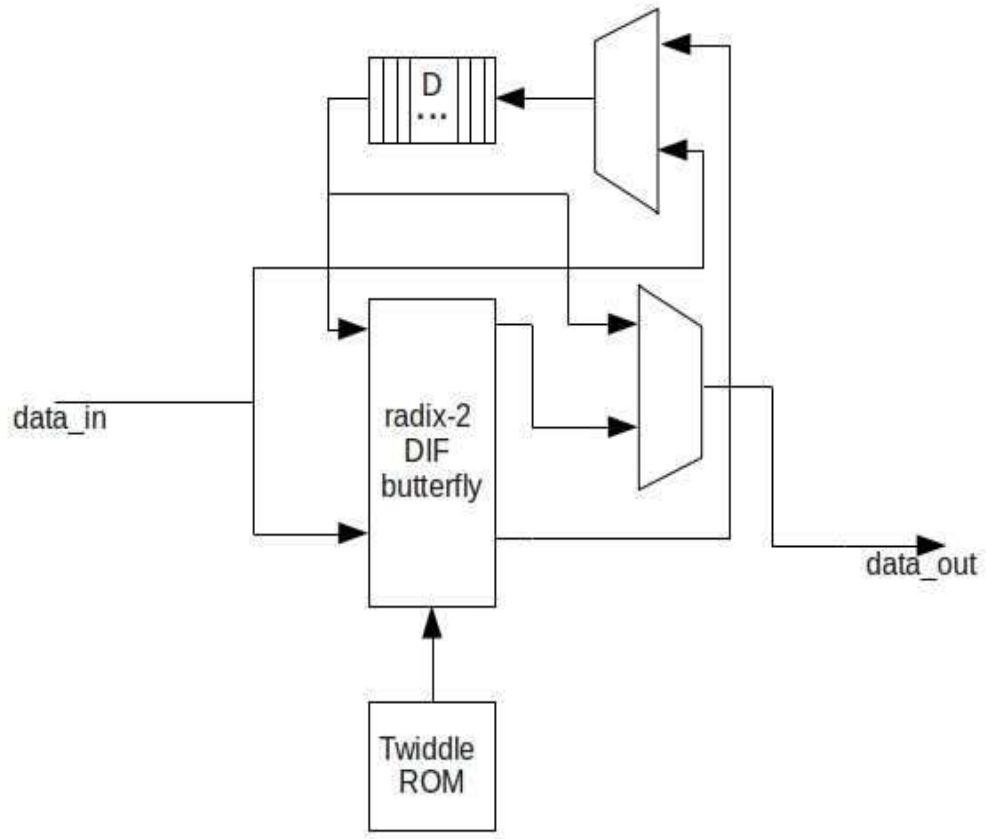


Figure 3.7 R2SDF stage block diagram

Variable point FFT can be achieved by adding Stages and in-between multiplexers. 1024, 512, 256, 128, or 64 point FFT Block Diagram is shown in figure below. Intermediate mux between stages decide the FFT length so FSM based controller is used to control the Data path.

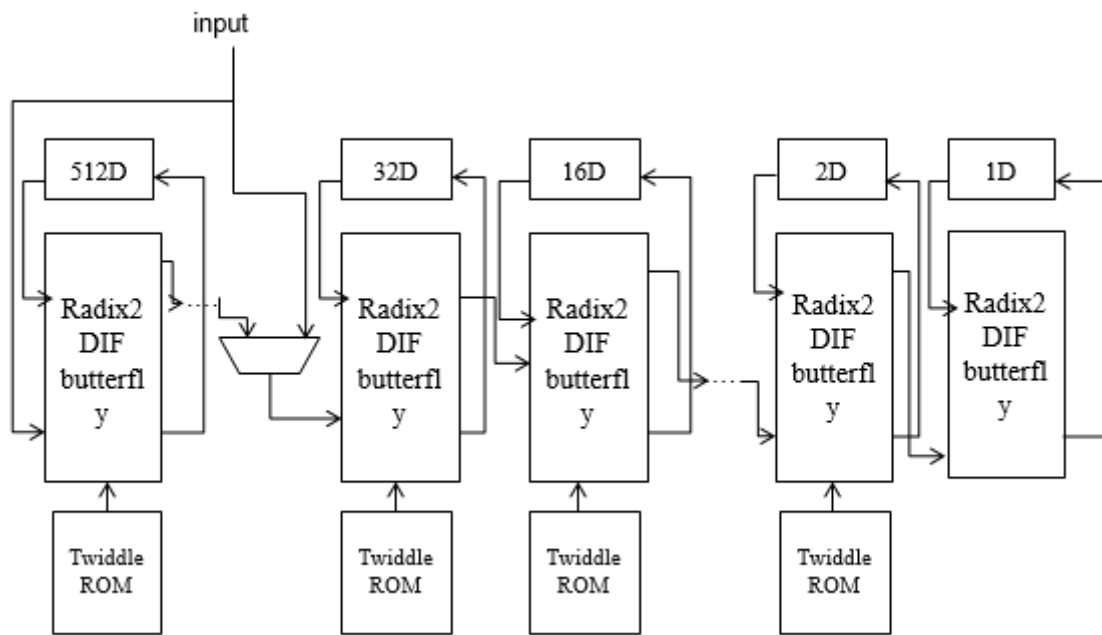


Figure 3.8 1024/512/256/128/64 point FFT Block Diagram for R2SDF algorithm

3.2.2 RTL block and simulation results

FFT has two block one is data path and other is controller, which based on FSM.

R2SDF pipelined FFT RTL block diagram shown below.

Timing Summary:

Speed Grade: -4

Minimum period: 130.380ns (Maximum Frequency: 7.670MHz)

Minimum input arrival time before clock: 121.725ns

Maximum output required time after clock: 14.050ns

Maximum combinational path delay: 14.479ns

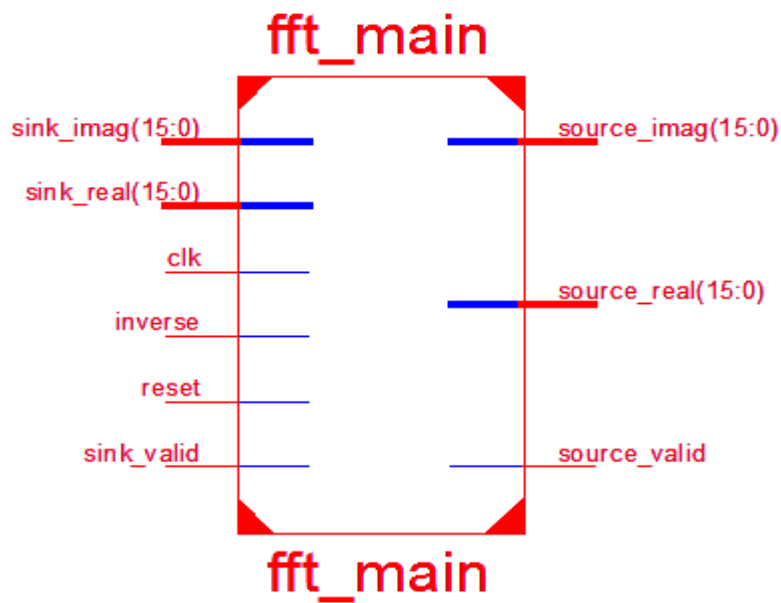


Figure 3.9 Top RTL block of 8-point FFT

Top FFT block have real and imaginary data input with 8-bit precision, clock and reset to synchronize input data, inverse input to find the IFFT, and sink_valid input to initiate the FFT, and Out_valid output tell the output start frame and end frame.

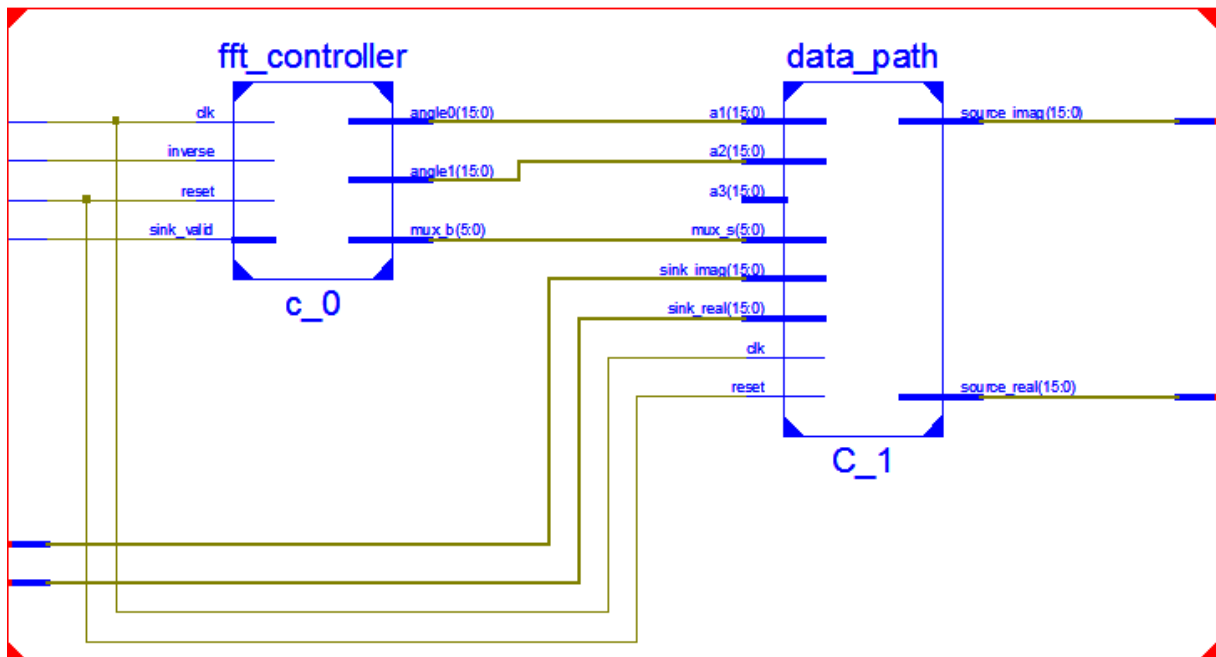


Figure 3.10 Controller and data path block diagram for 8-point FFT

FFT controller is designed by FSM, output of controller direct the data route in data path, and generate angle for Cordic to calculate twiddle factor. Each state has FIFO, radix-2 butterfly, Cordic and mux.

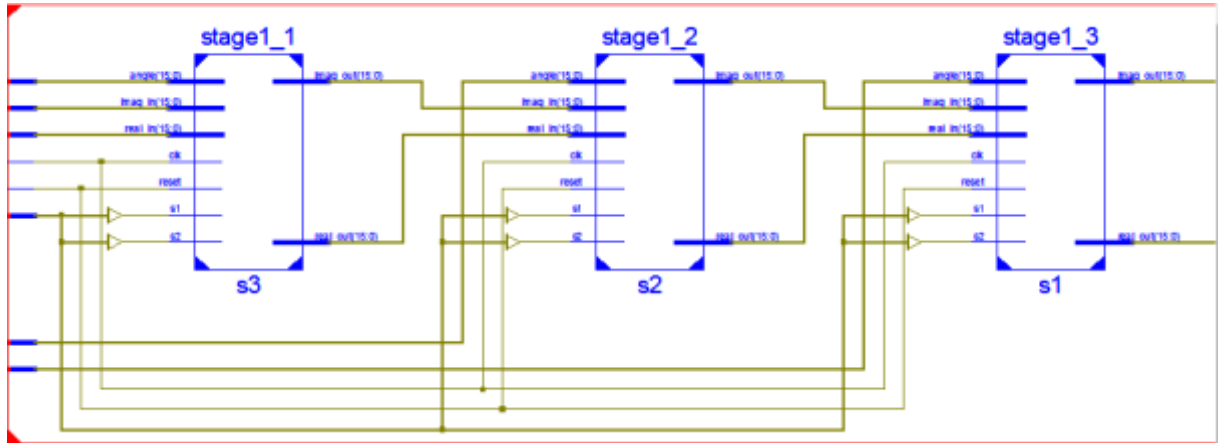


Figure 3.11 8-point FFT stage RTL Circuit

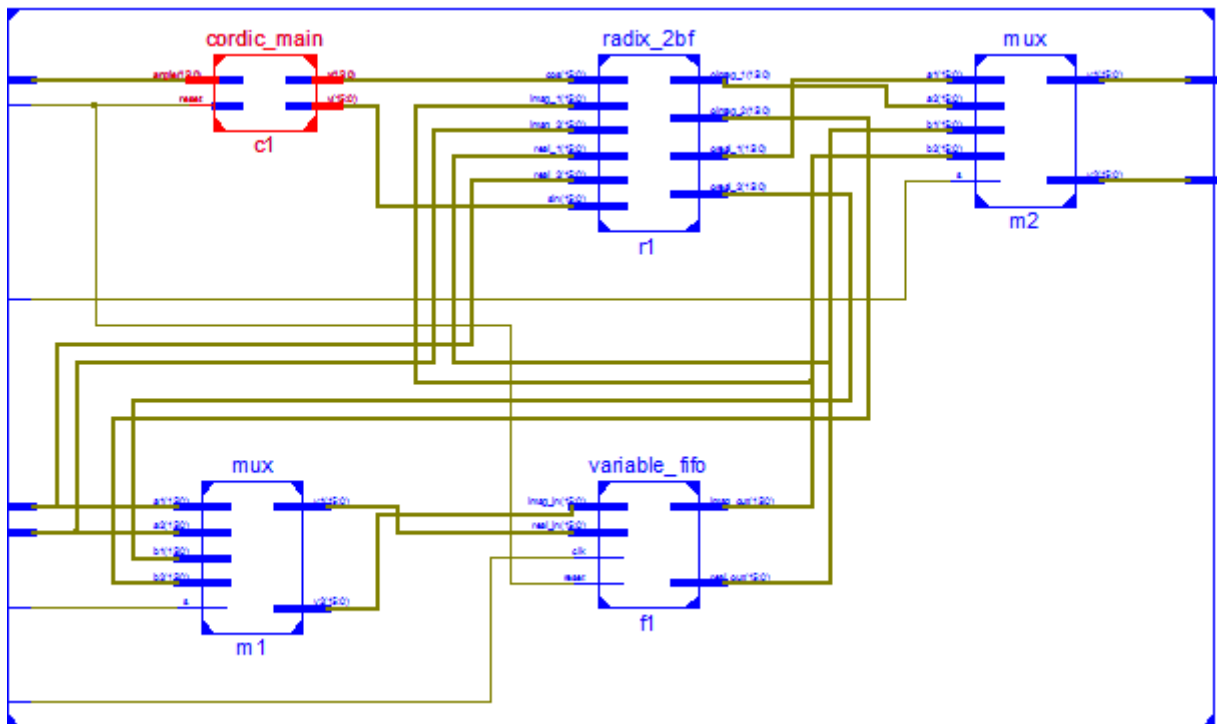


Figure 3.12 R2SDF FFT stage RTL circuit diagram

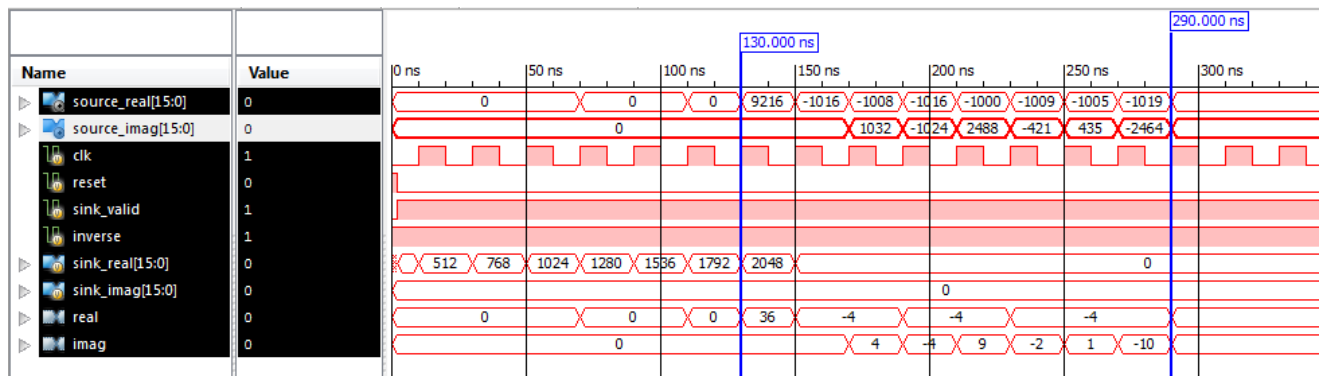


Figure 3.13 8-point FFT simulation Result

Output of FFT are in bit reversal mode

```
c =
Columns 1 through 7
36.0000 + 0.0000i -4.0000 + 9.6569i -4.0000 + 4.0000i -4.0000 + 1.6569i -4.0000 + 0.0000i -4.0000 - 1.6569i -4.0000 - 4.0000i

Column 8
-4.0000 - 9.6569i
```

Figure 3.14 MATLAB result of 8-point FFT

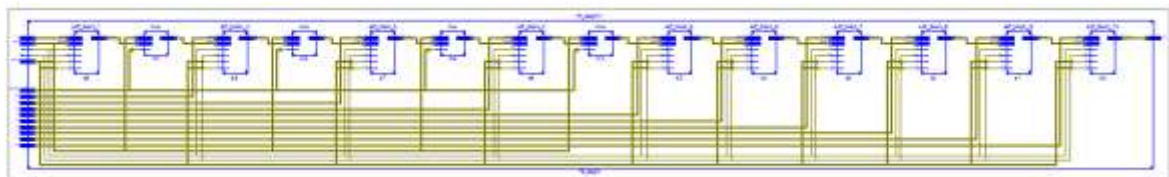


Figure 3.15 1024/512/256/.../64 point FFT RTL Datapath block

Table 3.1 variable FFT Macro Statistics

Macro Statistics	Description	Numbers
ROMs	16x16-bit ROM	150
Multiplier	17x17 bit multiplier	20
Multiplier	18x16 bit multiplier	10
Adders/Subtractors	15-bit adder	10
Adders/Subtractors	15-bit subtractor	10
Adders/Subtractors	16-bit adder carry out	10

Adders/Subtractors	16-bit addsub	40
Adders/Subtractors	16-bit addsub	470
Adders/Subtractors	17-bit Subtractors	30
Adders/Subtractors	18-bit Subtractors	10
Adders/Subtractors	23-bit Subtractors	10
Adders/Subtractors	34-bit adder	10
Register	Flip-Flops	33056
Register	17-bit comparator greater	20
Register	32-bit comparator less	10
logics	16-bit shifter	320

3.3 FFT in 2D

Suppose $x(n, m)$ is periodic with period N, M the analysis and synthesis formula of FFT given as

$$X(k, l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x(n, m) e^{-j2\pi(kn/N + lm/M)}$$

$$x(n, m) = \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} X(k, l) e^{j2\pi(kn/N + lm/M)}$$

2D FFT is calculated by 1D FFT first calculate FFT of all ROWs and then calculated FFT of all columns.

3.4 Implementation of Cordic Algorithm

CORDIC (Coordinate Rotation Digital Computer) is a well-known algorithm used to calculate trigonometry, exponential, hyperbolic function. CORDIC used only adder and shift register to calculate trigonometry function, no multipliers required. Cordic is a iterative algorithm in which fixed vector is rotated to calculate trigonometry functions.

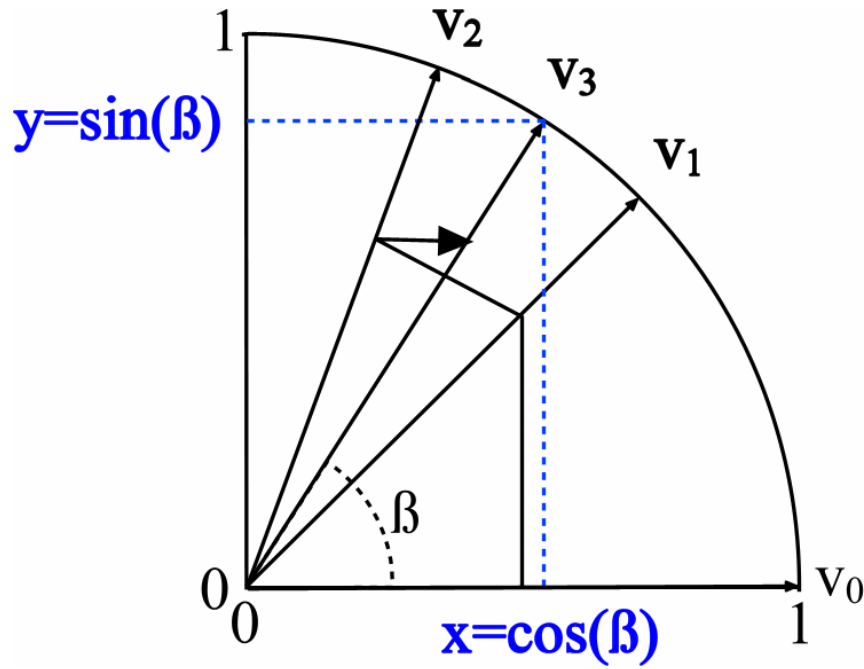


Figure 3.16 vector rotation in a unit radius circle

To determine sine and cosine of an angle β , the x and y coordinate value of vector V_3 on unit circle gives $\cos(\beta)$ and $\sin(\beta)$ respectively. In Cordic algorithm it start from vector V_0 . Vector V_0 successively rotating with fixed step size until the required angle has been achieved [15]. The rotation matrix given as follow

$$R_i = \begin{bmatrix} \cos\phi_i & -\sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix}$$

Iterative vector can be describe by multiplication of rotation matrix and previous vector define as

$$V_i = R_i V_{(i-1)}$$

$$\cos\alpha = \frac{1}{\sqrt{1 + \tan^2\alpha}}$$

$$\sin\alpha = \frac{\tan\alpha}{\sqrt{1 + \tan^2\alpha}}$$

If step angle of rotation is $\alpha_i = \arctan(2^i)$ then rotated vector give as

$$V_i = K_i \begin{bmatrix} 1 & -\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_{i-1} \\ y_{i-1} \end{bmatrix}$$

$$\text{Where } K_i = \frac{1}{\sqrt{1 + 2^{-2i}}}$$

Rotation can be clockwise or anticlockwise, which is decided by subtracting the step angle from desired angle.

$$\beta_i = \beta_{i-1} - \sigma_i \alpha_i$$

Cordic can be operated in two modes rotation mode or vector mode.

3.4.1 RTL block and simulation results

Table 3.2 HDL Synthesis Report Cordic macro Statistics

Macro Statistics	Description	Numbers
ROMs	32x32-bit ROM	19
Adders/Subtractors	32-bit adder	2
Adders/Subtractors	32-bit addsub	60
Multiplexers	32-bit 4-to-1	3
logics	32-bit shifter	40

Timing Summary:

Speed Grade: -4

- Maximum combinational path delay: 118.121ns

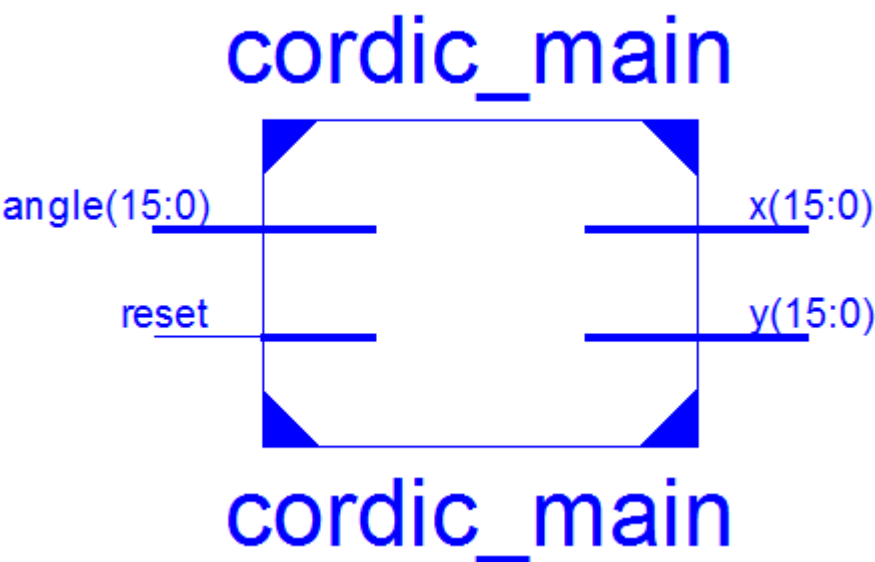


Figure 3.17 Cordic top level RTL block

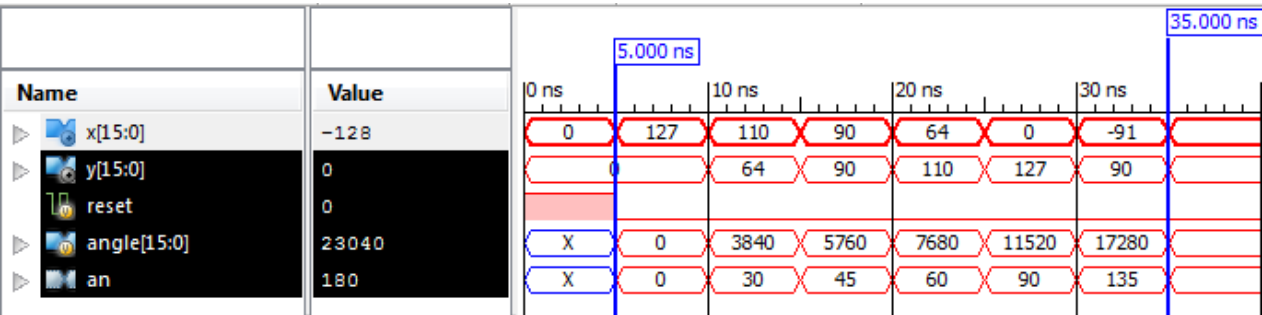


Figure 3.18 Cordic simulation result

3.5 Implementation of Gaussian kernel classifier

The main block of algorithm is Gaussian kernel classifier, it takes the input samples and train the classifier to find the response on test image. Kernel find the response at all subwindows by using FFT in a sliding manner. Gaussian kernel define as below equation

$$k^{gauss} = \exp\left(-\frac{1}{\sigma^2} (||x||^2 + ||x'||^2 - 2 F^{-1}(F(x) \Theta F(x')))\right)$$

Where Θ element wise multiplication, and F is is FFT

Calculation of exponential function Cordic is used define as follow

$$X_{i-1} = X_i - \sigma_i 2^{-i} y_i$$

$$Y_{i-1} = Y_i + \sigma_i 2^{-i} X_i$$

The above give equation are define Cordic equation for hyperbolic function calculation $X = \cosh$, $Y = \sinh$, and i is the no of iteration i varies from 1,2,3,4,4...

3.5.1 RTL block and simulation results

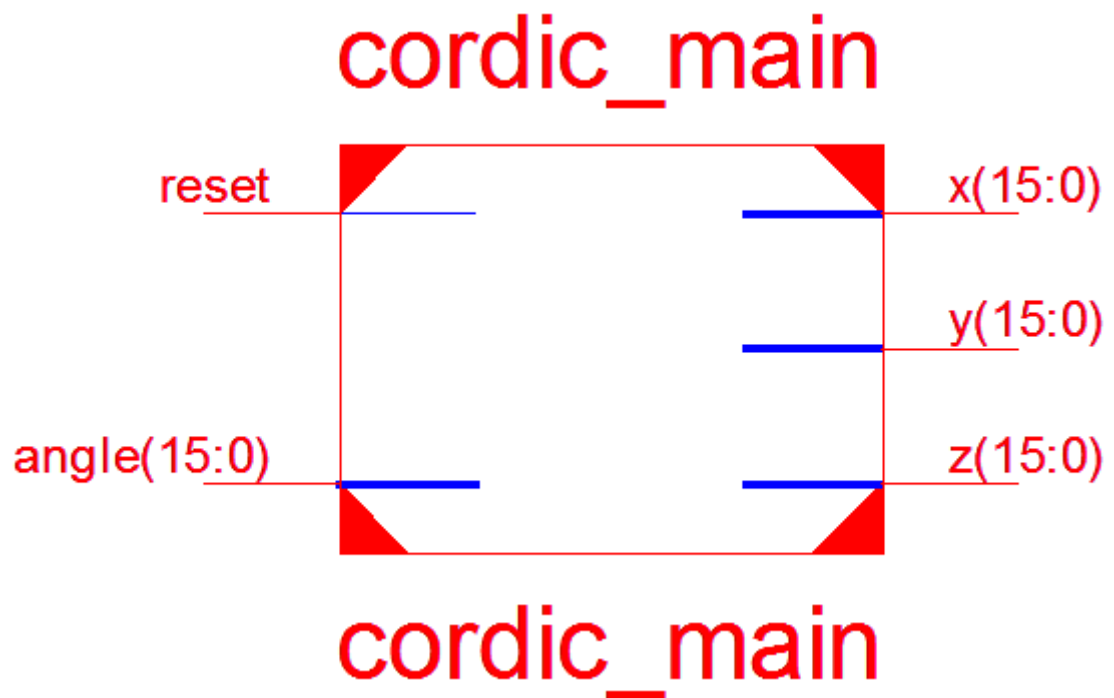


Figure 3.19 RTL Block of Cordic for Exponential function calculation

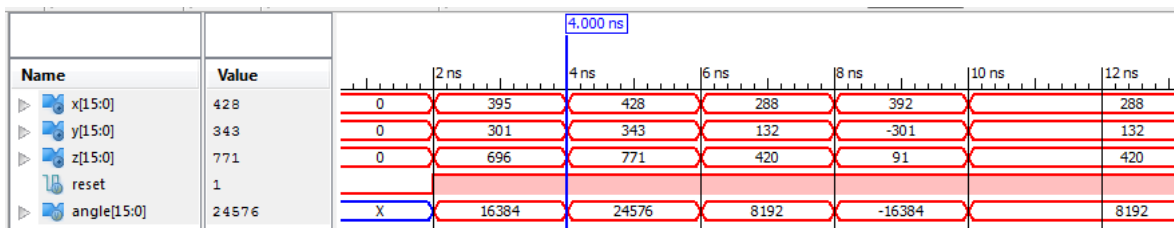


Figure 3.20 simulation result of Cordic for exponential function

3.6 Implementation of VGA Controller

VGA is acronym for Video Graphics Array, it is a video display standard. For standard VGA format a display have 640,480 columns and rows pixel, image is shown by turn off or on these pixel. VGA provide the interface between monitor and CPU

3.6.1 Timing diagram

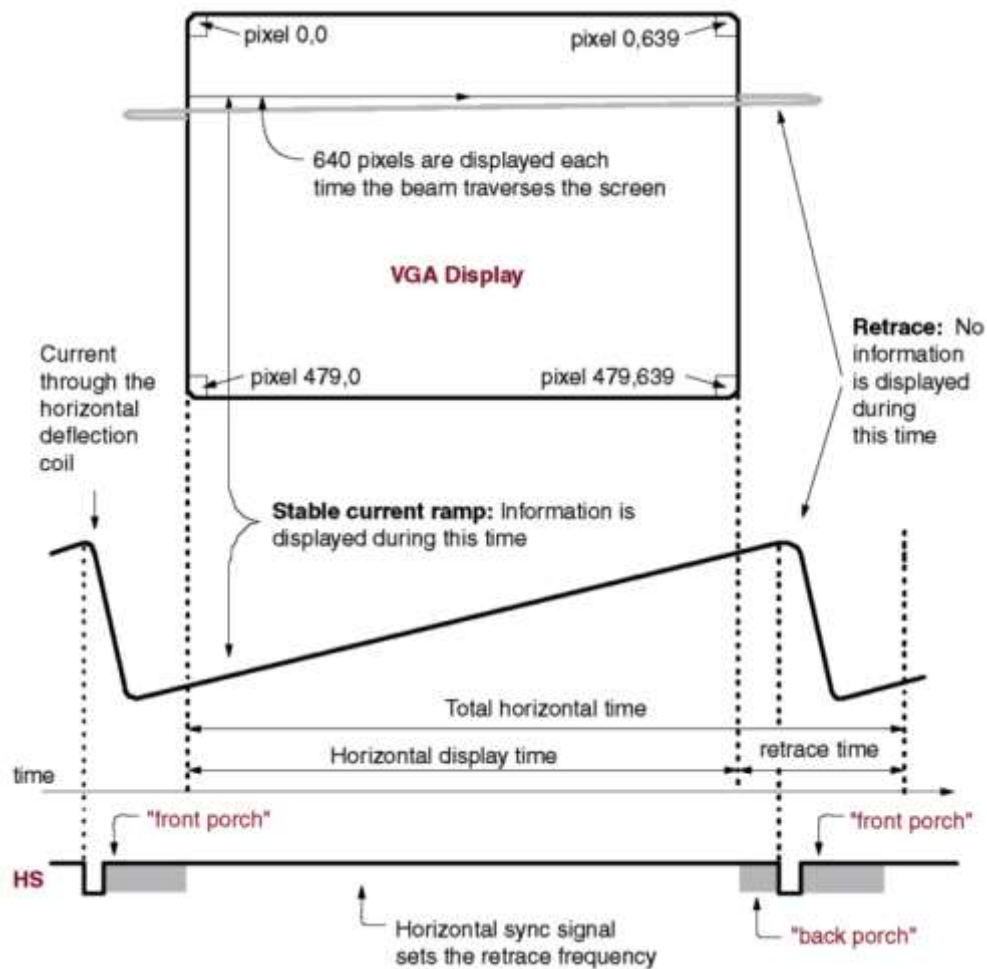


Figure 3.21 VGA timing diagram

3.6.2 RTL block and simulation results

Timing Summary:

Speed Grade: -4

Minimum period: 11.320ns (Maximum Frequency: 88.339MHz)

Minimum input arrival time before clock: 13.272ns

Maximum output required time after clock: 11.491ns

Maximum combinational path delay: 6.531ns

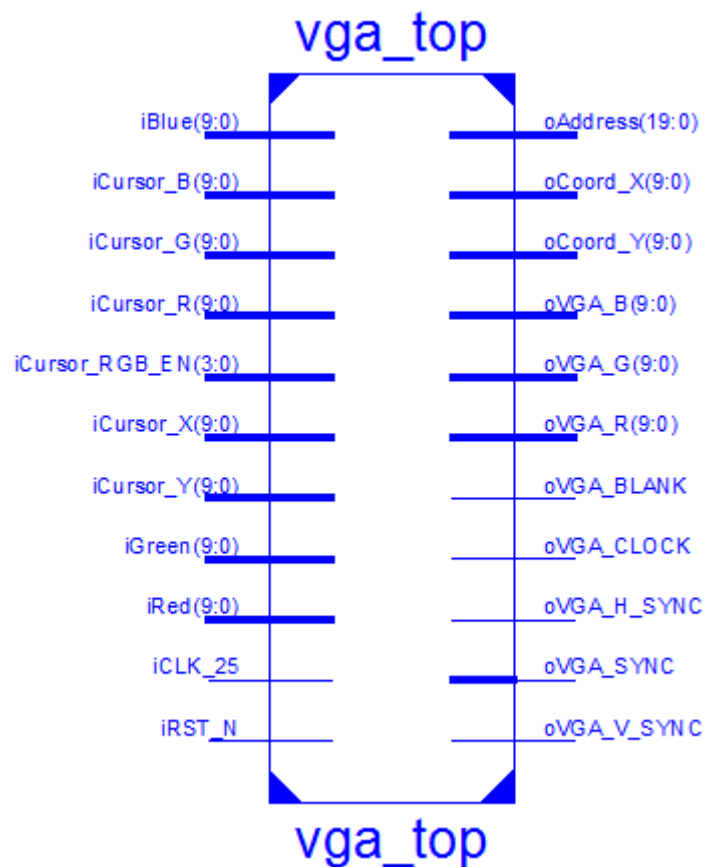


Figure 3.22 VGA Controller RTL Block

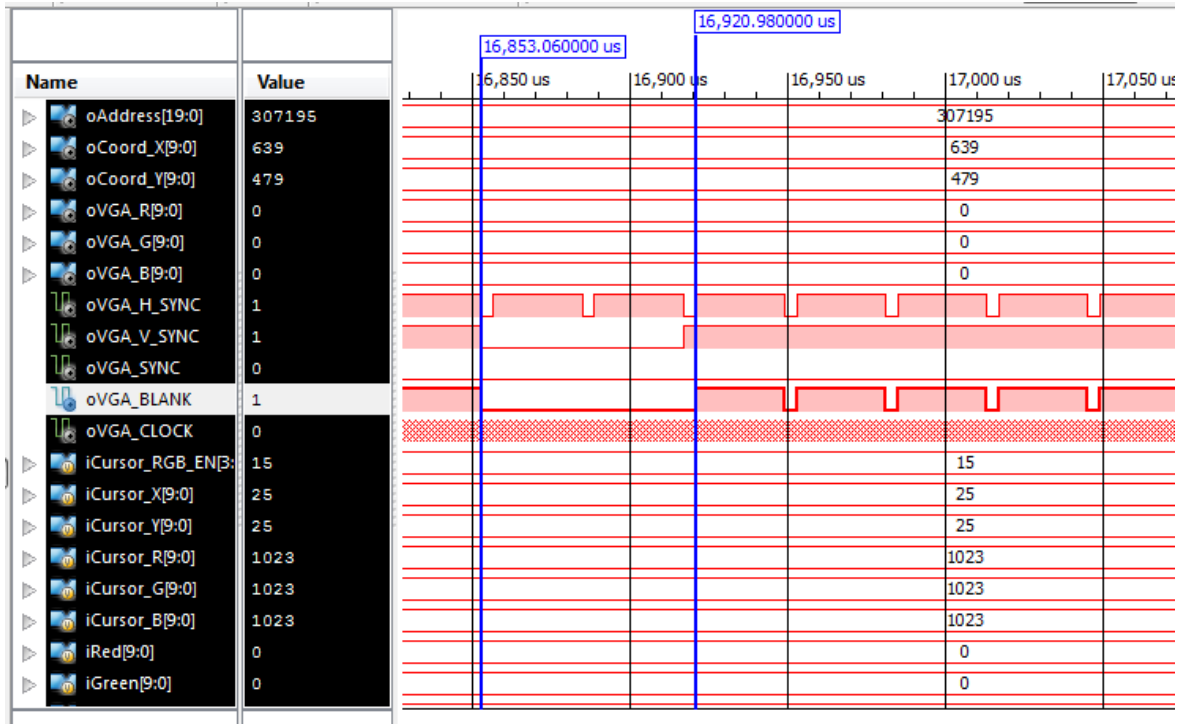


Figure 3.23 Simulation result of VGA Controller

Table 3.3 Device utilization summary of different modules

Module	Number of slice used		Number of LUTs		Number of IOBs		Minimum time period(ns)
	used	Utilization %	used	Utilization %	used	Utilization %	
8-pint FFT	974	20%	1933	20%	68	29%	130.380 ns
Cordic for trigonometry function	3	0%	5	0%	49	21%	80.606 ns
Cordic for exponential function	435	9%	856	9%	65	28%	83.592 ns
VGA controller	151	3%	275	2%	161	69%	11.320ns

The above utilization table is based on the Xilinx Spartan 3E device

3.7 Summary

All module of the algorithm implemented in Verilog HDL and simulation results compared with MATLAB. These modules are used for algorithm block, which is used with other interfacing module for complete embedded system which is describe in next chapter. Variable FFT define in this chapter is used most of the part of device FFT is streaming FFT which produce output of whole frame of an image. As shown other modules are simple and required very less area.

Chapter 4

Conclusion and Future Work

4.1 Conclusion

Object tracking system require high performance computer, for real time tracking in video processing system exploiting the parallelism of reconfigurable hardware. Image and video processing based algorithms, which implemented on FPGAs has several constraints like resource utilization, timing closures, power and speed. All the necessary hardware modules for tracking-by-detection algorithm are effectively implemented and verified. Tracking algorithm has many application in embedded system, for rapid prototyping of video processing algorithm used FPGA. These VLSI architectures are designed and verified successfully using Verilog HDL.

4.2 Future Work

To design tracking system the custom module are used with standard IP cores to interface with other systems. The system partitioned into software and hardware parts and optimize both for tracking in real time. Improve in algorithm for automatic object tracking and to detect the object if is out of plane.

Bibliography

- [1] Alper Yilmaz, Omar Javed, Mubarak Shah, “Object tracking: A Survey”, ACM Computer Survey 38, 4, Article 13, Dec. 2006
- [2] Himani S. Parekh, Darshak G. Thakore, Udesang K. Jaliya “A Survey on Object Detection and Tracking Methods”, IJIRCCE, vol 2, Issue 2, February 2014
- [3] J.Joshan Athanesious, P.Suresh, “Systematic Survey on Object Tracking Methods in Video”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), pp. 242-247, October 2012.
- [4] J. F. Henriques, R. Caseiro, Pedro Martins, and Jorge Batista, “Exploiting the Circulant Structure of Tracking-by-detection with Kernels”, ECCV 4, 2012
- [5] Simon J.D.Prince, “Computer Vision Models, Learning and Inference”, In: Cambridge University, pp. 81-212, 2012.
- [6] Alex Smola and S.V.M. Vishwanathan, “Introduction to Machine Learning”, In: Cambridge University Press, pp. 165-194, 2010.
- [7] Jean Gao, Akio Kosaka, Avinash C.Kak, “A multi-kalman filtering approach for video tracking of human-delineated objects in cluttered environments”, In Elsevier on Computer Vision and Image Understanding 102, pp. 260-316, 2006.
- [8] Sheikh Nasrullah, Dawood Ashraf Khan, “A Novel Alogorithm For Real Time Moving Object Tracking”, ICIIP 2011.
- [9] Jung UK Cho, Seung Hun Jin, Xuan Dia Pham, et al., “A Real-Time Color Feature Tracking Using Color Histograms”, Conference on Control, Automation and System 2007.

- [10] Dorin Comaniciu, Visvanathan Ramesh, “Kernel-Based Object Tracking”, IEEE Transection on Pattern Analysis and Machine Intelligence, VOL 25, No. 5, MAY 2003.
- [11] David S.Bolme, J.Ross Beveridge, et al., “Visual Object Tracking using Adaptive Correlation Filters”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2544-2550, June 2010.
- [12] Boris Babenko, Ming-Hsuan Yang, Serge Belongie, “Robust Object Tracking With Online Multiple Instance Learning”, IEEE Transaction on Pattern Analysis and Machine Intelligence, pp. 1619-1632, Dec 2010.
- [13] Huiyu Zhou, Yuan Yuan, Chunmei Shi, “Object tracking using SIFT features and mean shift”, In Elsevier on Computer Vision and Image Understanding 113, pp. 345-352, 2009.
- [14] Renan Netto, Pedro Michel, et al. “A Low-Power High Throughput Configurable FFT/IFFT Processor for WLAN and WiMax Protocols”, XXVII SIM- South Symposium on Microelectronics, 2012.
- [15] Kavya Sharat, Dr. B.V. Uma, Sagar D.M. “Calculation of Sine and Cosine of an Angle using the CORDIC Algorithm”, International Journal of Innovative Technology and Research Vol. 2, Issue 2, pp. 891-895 March 2014.